Analytical tracking along streamlines in temporally linear Raviart-Thomas velocity fields

T.F. Russell
University of Colorado at Denver, Denver, CO USA

R.W. Healy
U.S. Geological Survey, Denver, CO USA

ABSTRACT: Numerical simulators of ground-water flow and transport are frequently used to determine streamlines and to estimate travel times and pathways of contaminant movement. These items are often obtained by tracking conceptual water particles through the computational grid using model-calculated flow velocities. Such tracking is also an important component of the Lagrangian part of many methods for transport modeling, in particular the Eulerian-Lagrangian localized adjoint method (ELLAM). A procedure for exact analytical particle tracking is presented, given a lowest-order Raviart-Thomas velocity field $\mathbf{v}$ on a rectangular spatial grid, with linear temporal interpolation of $\mathbf{v}$ from the beginning to the end of a time step. This includes xt- and yt-bilinearity in the x- and y-components, respectively, of $\mathbf{v}$. Previous authors assumed that $\mathbf{v}$ was steady, or that its time derivative was constant in space. Transience in $\mathbf{v}$ allows a particle to reverse its direction during a time step. The added effect of bilinearity can be significant, especially when $\mathbf{v}$ varies in time due to changes in well pumping rates or variable recharge. These effects are discussed qualitatively and illustrated with test problems that compare the accuracy of the tracking methods.

1 INTRODUCTION

Problems of ground-water flow and solute transport have been intensely studied in recent years due to increased awareness of the susceptibility of ground water to contamination. Numerical simulators of ground-water flow and solute transport are common tools employed in these analyses. Numerical models are frequently used to determine streamlines and estimate travel times and pathways of contaminant movement. These items are often obtained by tracking imaginary particles of water through the model's computational grid using model-calculated flow velocities. Pollock (1988, 1989) used particle tracking to determine the location of streamlines and to estimate the time required for water to traverse different segments of a model domain. Tracking was also applied to estimate the extent of advective transport of solutes (Garabedian & Konikow 1983). Other uses include its incorporation into numerical models of solute transport that are based on the advection-dispersion equation (Konikow & Bredehoeft 1988; Prickett et al. 1981; Zheng 1989; Healy & Russell 1993). This paper

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
2006		**00-00-2006 to 00-00-2006**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Analytical tracking along streamlines in temporally linear Raviart-Thomas velocity fields	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **University of Colorado at Denver,Center for Computational Mathematics,PO Box 173364,Denver,CO,80217-3364**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT
see report

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**14**	

presents a new method for particle tracking.

The first step in a general particle tracking procedure is to solve the ground-water flow equation using standard finite-difference or finite-element space and time grids to obtain estimates of hydraulic head or pressure at fixed node locations and times. Darcy's equation is then employed to calculate velocities from the head or pressure field. Alternatively, heads and velocities can be solved simultaneously as in the method of mixed finite elements (Raviart & Thomas 1977; Russell & Wheeler 1983). Either approach results in velocities that are calculated at the interfaces between adjacent nodes. These velocities are then used to calculate the pathlines and travel times of particles.

We assume a Raviart-Thomas (1977) velocity field: within each spatial cell, the x-component is continuous piecewise-linear in the x-direction and discontinuous piecewise-constant in the y-direction, and the y-component is the reverse. This was also assumed by Pollock (1988, 1989) and Schafer-Perini & Wilson (1991). This vector field lends itself naturally to linear interpolation for determining velocities in the interior of each cell. Linear interpolation, however, does produce discontinuities in the velocity field at cell interfaces (Goode 1990).

Calculation of travel path lines for any particle requires integration of the particle velocity over time. This integration can be accomplished by one of three approaches: analytical, numerical, or semianalytical. The analytical approach (Javandel et al. 1984) produces exact solutions, but only for a limited number of ideal cases of steady flow, homogeneous media, and simple geometry. Such cases seldom arise in practice.

Numerical techniques for integration include the explicit single-step method (Goode 1990), the first-order Euler (Lu 1994), and the fourth-order Runge-Kutta (Nelson 1978; Shafer 1987; Zheng 1989). These schemes are not limited by transient velocities or complexity in the velocity fields. The single-step explicit method is computationally simple, but of limited accuracy. The other two methods can attain a high degree of accuracy, but may require a large number of steps (and therefore computation time) to do so.

The semianalytical technique combines aspects of analytical and numerical methods. It makes use of an analytical solution to the integral within an individual space and time cell under the assumption of the Raviart-Thomas velocity field. Tracking is then conducted through one cell at a time. This idea was first presented by Pollock (1988, 1989) for steady-state flow conditions and has since been used by Schafer-Perini & Wilson (1991) and Healy & Russell (1993). It was extended to include nonsteady-state conditions under the added assumption that velocities varied linearly with time within each time step and that the spatial derivative of velocity was constant in time (i.e., $\partial^2 v/\partial x \partial t = 0$) (Lu 1994).

In this paper we extend the work of Lu (1994) and obtain the tracking equations for nonzero $\partial^2 v/\partial x \partial t$. Test problems are presented to demonstrate the sensitivity of results to inclusion of this term. Because of the complex nature of the tracking equations, a detailed algorithm is provided. The algorithm permits straightforward treatment of single or double reversals in velocity direction that may occur within a single time step.

2 TRACKING EQUATIONS

2.1 *Decoupled ordinary differential equations*

Let $C = [x_{i-1}, x_i] \times [y_{j-1}, y_j] \times [t^n, t^{n+1}]$ be a space-time cell in a two-dimensional rectangular grid over a time step. The developments described below will generalize in an obvious way

to three space dimensions. In C, suppose that a particle is located at (x_0, y_0) at time t_0. Let

$$\mathbf{v}_0 = (v_0^x, v_0^y) = (v^x(x_0, y_0, t_0), v^y(x_0, y_0, t_0)) \tag{1}$$

be the velocity vector at this location and time. Spatially, the velocity belongs to the lowest-order Raviart-Thomas space RT_0, and the RT_0 coefficients are linear functions of time, so that v^x is bilinear in x and t and constant in y, and v^y is bilinear in y and t and constant in x. Accordingly, if we let

$$v_x = \frac{\partial v^x}{\partial x}(t_0) \quad \text{(independent of } x \text{ and } y\text{)}, \tag{2}$$

$$v_y = \frac{\partial v^y}{\partial y}(t_0) \quad \text{(independent of } x \text{ and } y\text{)}, \tag{3}$$

$$v_t^x = \frac{\partial v^x}{\partial t}(x_0) \quad \text{(independent of } y \text{ and } t\text{)}, \tag{4}$$

$$v_t^y = \frac{\partial v^y}{\partial t}(y_0) \quad \text{(independent of } x \text{ and } t\text{)}, \tag{5}$$

$$v_{xt} = \frac{\partial^2 v^x}{\partial x \partial t} \quad \text{(constant)}, \tag{6}$$

$$v_{yt} = \frac{\partial^2 v^y}{\partial y \partial t} \quad \text{(constant)}, \tag{7}$$

then we can write the velocity components on C as

$$v^x(x,t) = v_0^x + v_t^x \Delta t + (v_x + v_{xt}\Delta t)\Delta x, \tag{8}$$

$$v^y(y,t) = v_0^y + v_t^y \Delta t + (v_y + v_{yt}\Delta t)\Delta y, \tag{9}$$

where $\Delta x = x - x_0$ and $\Delta y = y - y_0$.

The tracking of the particle across C, which amounts to determining the trajectory $(x(t), y(t))$, is governed by the ordinary differential equations

$$x'(t) = v^x(x(t), t), \qquad x(t_0) = x_0, \tag{10}$$

$$y'(t) = v^y(y(t), t), \qquad y(t_0) = y_0. \tag{11}$$

Because v^x does not depend on y and v^y does not depend on x, these ODE's are decoupled, and each can be solved on C without reference to the other. The analogous observation holds in three dimensions.

2.2 *Global tracking algorithm*

The global tracking algorithm consists of a sequence of local tracking steps, each of which is confined to a single space-time cell C. A step begins at an initial time $t_0 \geq t^n$, which is

either the starting time for the trajectory or the time at which the trajectory enters C. The step lasts until a final time $t_f \leq t^{n+1}$, which is either the ending time for the trajectory or the time at which it leaves C (note that it leaves at time t^{n+1} if it does not cross the spatial boundary of C).

Because the ODE's (10) and (11) are decoupled, one can separately determine times $t_x > t_0$ and $t_y > t_0$ at which $x(t)$ and $y(t)$, respectively, leave the ranges corresponding to C. Then t_f is set equal to the minimum, $t_f = \min\{t_x, t_y\}$. Similarly, in three dimensions, t_f is the minimum of three times determined by independent one-dimensional calculations. Thus, the objective of the detailed algorithm below is to carry out one local step in one dimension. These local one-dimensional steps can be compared to find t_f, then concatenated to follow the global trajectory.

2.3 *Global tracking equations*

Accordingly, we confine ourselves henceforth to a single dimension x. The ODE (10), with terms defined in (2), (4), (6), and (8), can be solved analytically. We drop the superscript x from v_0^x and v_t^x in (1), (4), and (8). Suppose for now that these definitions hold globally, i.e., ignore the fact that they are different on cells other than C. The analytical solution is

$$\begin{aligned} x(t) = x_0 &+ \frac{v_t}{v_{xt}}\left(e^{\frac{1}{2}v_{xt}\Delta t^2 + v_x \Delta t} - 1\right) \\ &+ \left(v_0 - \frac{v_x v_t}{v_{xt}}\right) e^{\frac{1}{2}v_{xt}\left(\Delta t + \frac{v_x}{v_{xt}}\right)^2} \sqrt{\frac{\pi}{2v_{xt}}} \times \\ &\times \left[\operatorname{erfc}\left(\sqrt{\frac{v_{xt}}{2}}\frac{v_x}{v_{xt}}\right) - \operatorname{erfc}\left(\sqrt{\frac{v_{xt}}{2}}\left(\Delta t + \frac{v_x}{v_{xt}}\right)\right)\right], \\ &\qquad\qquad v_{xt} > 0, \end{aligned} \tag{12}$$

$$\begin{aligned} x(t) = x_0 &+ \frac{v_t}{v_{xt}}\left(e^{\frac{1}{2}v_{xt}\Delta t^2 + v_x \Delta t} - 1\right) \\ &+ \left(v_0 - \frac{v_x v_t}{v_{xt}}\right) e^{\frac{1}{2}v_{xt}\left(\Delta t + \frac{v_x}{v_{xt}}\right)^2} \sqrt{\frac{2}{-v_{xt}}} \times \\ &\times \int_{\sqrt{\frac{-v_{xt}}{2}}\frac{v_x}{v_{xt}}}^{\sqrt{\frac{-v_{xt}}{2}}\left(\Delta t + \frac{v_x}{v_{xt}}\right)} e^{s^2}\, ds, \qquad v_{xt} < 0, \end{aligned} \tag{13}$$

$$x(t) = x_0 + v_0 \frac{e^{v_x \Delta t} - 1}{v_x} + v_t \frac{e^{v_x \Delta t} - (1 + v_x \Delta t)}{v_x^2}, \qquad v_{xt} = 0, \quad v_x \neq 0, \tag{14}$$

$$x(t) = x_0 + v_0 \Delta t + \frac{1}{2} v_t \Delta t^2, \qquad v_{xt} = v_x = 0. \tag{15}$$

Equations (14) and (15) are those of Lu (1994). In our more general context, (12) through (15) are not satisfactory for a practical implementation, due to difficulties with roundoff error. Small values of v_{xt} require an alternative equation that eases the transition from (12) or (13) to (14). Let ϵ denote the machine precision, i.e., the smallest positive number such that $1+\epsilon > 1$. Analysis of (12) and (13) finds roundoff error $O(v_{xt}^{-1}\epsilon)$. Then asymptotic techniques to determine the behavior of $x(t)$ for small v_{xt} lead to (14), plus a first-order term in v_{xt}, with truncation error $O(v_{xt}^2)$. As v_{xt} tends to 0, the roundoff error increases and the truncation error decreases, yielding a threshold $|v_{xt}| = \delta$ above which (12) or (13) is used, and below which (14) with the additional first-order term is used. The threshold is where the two errors are equal, $O(v_{xt}^{-1}\epsilon) = O(v_{xt}^2)$, so that $\delta = O(\epsilon^{1/3})$ and the error is $O(\epsilon^{2/3})$. The actual implemented equations are

$$x(t) = \text{right-hand side of (12)}, \qquad v_{xt} \geq \delta, \tag{16}$$

$$x(t) = \text{right-hand side of (13)}, \qquad v_{xt} \leq -\delta, \tag{17}$$

$$\begin{aligned} x(t) = x_0 + v_0\frac{e^{v_x\Delta t}-1}{v_x} + v_t\frac{e^{v_x\Delta t}-(1+v_x\Delta t)}{v_x^2} \\ + v_{xt}\left[\frac{v_0}{v_x}\left(\frac{1}{2}\Delta t^2 e^{v_x\Delta t} - \frac{e^{v_x\Delta t}-(1+v_x\Delta t)}{v_x^2}\right)\right. \\ + v_t\left(\frac{1}{2}\Delta t^2\frac{e^{v_x\Delta t}-(1+v_x\Delta t)}{v_x^2}\right. \\ \left.\left. - 3\frac{e^{v_x\Delta t}-(1+v_x\Delta t+\frac{1}{2}(v_x\Delta t)^2+\frac{1}{6}(v_x\Delta t)^3)}{v_x^4}\right)\right], \end{aligned}$$

$$|v_{xt}| < \delta, \qquad v_x \text{ not small}, \tag{18}$$

where

$$\begin{aligned} \delta = \left|2\epsilon v_t \max\{1, e^{v_x\Delta t}\}\right|^{1/3} \times \\ \times\left(\left|\frac{v_t v_x \Delta t^4}{8}\frac{e^{v_x\Delta t}-(1+v_x\Delta t+\frac{1}{2}(v_x\Delta t)^2)}{v_x^3}\right|\right. \\ \left. + \left|\frac{v_0\Delta t^4}{8}\frac{e^{v_x\Delta t}-1}{v_x}\right| + \left|\frac{v_t\Delta t^6}{48}\right|\right)^{-1/3}. \end{aligned} \tag{19}$$

In (18) and (19), if v_x is small, make the following replacements, letting $s = v_x \Delta t$:

Replace	with	if $\lvert s \rvert <$
$\frac{e^s - 1}{v_x}$	$\Delta t(1 + \frac{s}{2})$	$(6\epsilon)^{1/3}$
$\frac{e^s - (1+s)}{v_x^2}$	$\Delta t^2(\frac{1}{2} + \frac{s}{6})$	$(24\epsilon)^{1/4}$
$\frac{1}{v_x}\left(\frac{1}{2}\Delta t^2 e^s - \frac{e^s-(1+s)}{v_x^2}\right)$	$\Delta t^3(\frac{1}{3} + \frac{5s}{24})$	$(\frac{40}{3}\epsilon)^{1/5}$
$\frac{e^s - (1+s+\frac{1}{2}s^2+\frac{1}{6}s^3)}{v_x^4}$	$\Delta t^4(\frac{1}{24} + \frac{s}{120})$	$(720\epsilon)^{1/6}$
$\frac{e^s - (1+s+\frac{1}{2}s^2)}{v_x^3}$	$\Delta t^3(\frac{1}{6} + \frac{s}{24})$	$(120\epsilon)^{1/5}$

The next section discusses how these equations are used within the confines of a space-time cell C.

3 LOCAL TRACKING ALGORITHM

A complication in determining the time t_f when a trajectory leaves a cell is the possibility of reversals in the direction of the trajectory. Thus, one cannot simply assume that if the trajectory is within the range of C at the initial and final times, then it must be within the range at all intermediate times. This section analyzes the possibilities for reversals and then bases a detailed algorithm on the results.

3.1 *Reversals of direction*

Equation (8) for the velocity $v(x,t) = v^x(x,t)$ can be rewritten in the form

$$v(x,t) = v_{xt}\left(\Delta x + \frac{v_t}{v_{xt}}\right)\left(\Delta t + \frac{v_x}{v_{xt}}\right) + \left(v_0 - \frac{v_x v_t}{v_{xt}}\right), \qquad v_{xt} \neq 0, \tag{20}$$

$$v(x,t) = v_x \Delta x + v_t \Delta t + v_0, \qquad v_{xt} = 0, \tag{21}$$

where $x = x_0 + \Delta x$, $t = t_0 + \Delta t$, and we think of Δx and Δt as the variables. Reversals can occur if $v = 0$ somewhere in C, so we concern ourselves with the locus of points in the $(\Delta x, \Delta t)$-plane where v vanishes.

If $v_{xt} = 0$, then by (21) this locus is a straight line in the plane. If this line meets C and has negative slope (v_x and v_t have the same sign), the two essentially distinct reversal situations are depicted in Fig. 1. Each box represents the space-time cell C in the xt-plane (x on the horizontal axis), the solid line is the locus of $v = 0$, and the dashed line is a possible trajectory with slope $dt/dx = 1/v$ at each point. In the left figure, $v_x < 0$ and $v_t < 0$, so that $v > 0$ in the lower left and $v < 0$ in the upper right. The reverse is true in the right figure. Note that the trajectory can only cross the solid line vertically ($v = 0$ implies that the slope $1/v$ is infinite). If the locus has positive slope, mirror images of Fig. 1, with the signs of v reversed, cover the possibilities. In all cases, there is at most one reversal, and its presence can be detected by the initial and final velocities v_0 and v_f having opposite signs. These possibilities arise in Lu (1994). As a special case of (21), if the velocity is steady ($v_t = 0$), then the solid line in Fig. 1 would be vertical, and the trajectory could not cross it;

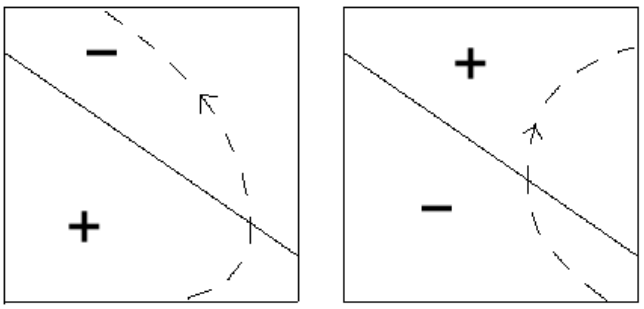

Figure 1: Reversal possibilities with linear velocity function.

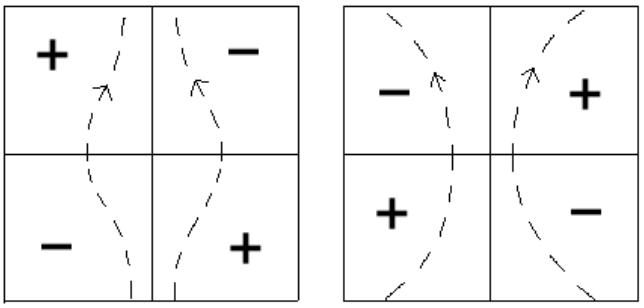

Figure 2: Reversal possibilities with bilinear velocity function, $v = 0$ on intersecting lines.

in other words, reversals cannot occur in a steady velocity field, as in Pollock (1988, 1989) and Schafer-Perini & Wilson (1991).

Next, suppose that $v_{xt} \neq 0$, so that (20) applies. The locus of $v = 0$ is a hyperbola with vertical asymptote $\Delta x = -v_t/v_{xt}$ and horizontal asymptote $\Delta t = -v_x/v_{xt}$, unless $v_0 v_{xt} = v_x v_t$, in which case the locus consists of the two asymptotes. The latter case is depicted in Fig. 2, with $v_{xt} < 0$ in the left figure and $v_{xt} > 0$ in the right. In either situation, at most one reversal is possible, and the signs of the initial and final velocities v_0 and v_f will flag it.

Now suppose that $v_0 v_{xt} < v_x v_t$, so that the locus is a hyperbola whose branches lie in the first and third quadrants relative to the asymptotes and have negative slope. The locus divides the plane into three regions. The possibilities can be classified according to the sign of v_{xt}, which determines the sign of v in each region, and according to which regions the lower-left and upper-right corners of C lie in. The six essentially distinct cases are shown in Fig. 3, with $v_{xt} < 0$ in the three left figures and $v_{xt} > 0$ on the right. In all but one of the cases, at most one reversal is possible, and it can be detected as above. The possibility of two reversals arises in the middle-left picture in Fig. 3; however, it can be flagged by the fact that the initial and final velocities are negative ($v_0 < 0$, $v_f < 0$), while the net movement from the initial to the final position is in the positive (opposite) direction ($x_f > x_0$). The cases for $v_0 v_{xt} > v_x v_t$ are covered by mirror images of Fig. 3 with the signs of v reversed.

The reversal cases can be summarized as follows. At most two reversals are possible. If there are two, then $x_f - x_0$ has sign opposite to that of v_0 and v_f, and conversely. If there is one, then v_0 and v_f have opposite signs, and conversely.

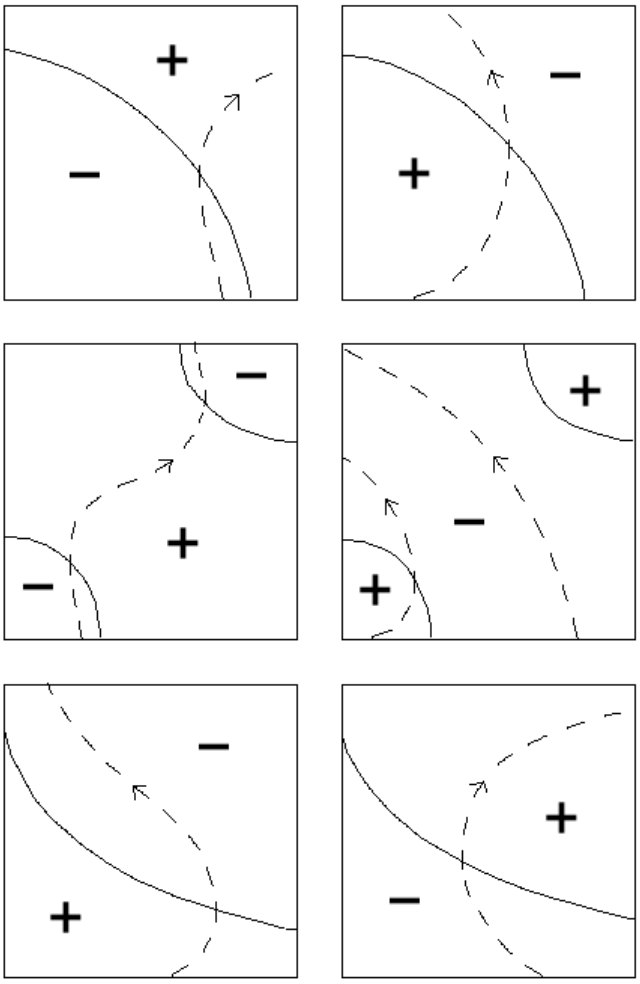

Figure 3: Reversal possibilities with bilinear velocity function, $v = 0$ on hyperbola.

3.2 *Detailed local tracking algorithm*

The algorithm described here is based on the analytical equations and the reversal possibilities derived above. The objective is to detect the earliest time at which a trajectory leaves the space-time cell C. If there is no reversal during the time step, and the trajectory crosses the cell boundary, this leaving time is found by solving the nonlinear equation $x(t) =$ boundary coordinate, where $x(t)$ is given by (16), (17), or (18). The solution is obtained by Newton's method, using (8) for the derivative $x'(t) = v^x(x(t), t)$, with a bisection method as a backup if a Newton step fails to make $x(t)$ closer to the desired value. If there is a reversal, then its time t^* is found by solving $v(x(t), t) = 0$, with v given by (8). Again Newton/bisection is used, with derivative

$$v'(t) = v_t + v_{xt}\Delta x + (v_x + v_{xt}\Delta t)v(t). \tag{22}$$

Once t^* is found, it is known that there are no reversals between t_0 and t^*, so that the time interval up to t^* can be treated as a step without reversals.

In the following step-by-step description, t_0 is the current time, at which the local tracking step begins; $x_0 = x(t_0)$ is the current position, from (16) through (18); v_0 is the current

velocity, from (1); v_x, v_t, and v_{xt}, from (2), (4), and (6), are the current parameters in the velocity equation (8), and may depend on the velocity direction if x_0 is at a cell boundary (choose those corresponding to the cell being entered by the trajectory); t_f is the end of the current time step, the goal of the local tracking step; and x_L and x_R are the left and right endpoints of the cell containing x_0 (the cell being entered by the trajectory if x_0 is at a cell boundary). Note that the "cell being entered" is unambiguous, because v is continuous at cell boundaries.

At the start of each local tracking step, t_0, x_0, v_0, v_x, v_t, v_{xt}, t_f, x_L, and x_R are set. To complete the step and be ready for the next step, the following algorithm determines new values for these parameters.

1. Compute the final position $x_f = x(t_f)$ from (16) through (18) and the final velocity $v_f = v(x_f, t_f)$ from (8), as if the current velocity parameters were global.

2. If v_0 and v_f have opposite signs, go to step 6. (If $v_0 = 0$, assign the sign of v_t; if $v_f = 0$, assign the sign of $-\dfrac{\partial v}{\partial t}(x_f, t_f) = -(v_t + v_{xt}(x_f - x_0))$.)

3. [v_0 and v_f have the same sign] If v_0 and v_f are negative, go to step 5.

4. [v_0 and v_f are positive] If $x_0 \le x_f < x_R$, go to step 15. If $x_R < x_f$, set $t_1 = t_0$, $t_2 = t_f$ and go to step 16. If $x_f = x_R$, go to step 18. Otherwise ($x_f < x_0$), go to step 9.

5. [v_0 and v_f are negative] If $x_L < x_f \le x_0$, go to step 15. If $x_f < x_L$, set $t_1 = t_0$, $t_2 = t_f$ and go to step 17. If $x_f = x_L$, go to step 20. Otherwise ($x_0 < x_f$), go to step 12.

6. [there is one reversal] Set $t_1 = t_0$, $t_2 = t_f$, and use Newton/bisection with (22) to find the unique t^*, $t_1 < t^* < t_2$, such that $v(x(t^*), t^*) = 0$. Evaluate $x^* = x(t^*)$.

7. If $x^* < x_L$, set $t_1 = t_0$, $t_2 = t^*$, and go to step 17. If $x_R < x^*$, set $t_1 = t_0$, $t_2 = t^*$, and go to step 16.

8. [reversal occurs within current cell] If $x_f < x_L$, set $t_1 = t^*$, $t_2 = t_f$, and go to step 17. If $x_R < x_f$, set $t_1 = t^*$, $t_2 = t_f$, and go to step 16. If $x_L < x_f < x_R$, go to step 15. If $x_f = x_L$, go to step 20. Otherwise ($x_f = x_R$), go to step 18.

9. [two reversals, v_0 and v_f positive, net movement to left] Set $t_1 = t_0$, $t_2 = t_f$, and use Newton/bisection with (8) to find the unique $\tilde{t}$, $t_1 < \tilde{t} < t_2$, such that $x(\tilde{t}) = (x_0 + x_f)/2$. Then set $t_1 = t_0$, $t_2 = \tilde{t}$, and use Newton/bisection with (22) to find the unique t^*, $t_1 < t^* < t_2$, with $v(x(t^*), t^*) = 0$; this is the time of the first reversal. Set $x^* = x(t^*)$.

10. If $x_R < x^*$, set $t_1 = t_0$, $t_2 = t^*$, and go to step 16.

11. [first reversal occurs within current cell] If $x(\tilde{t}) \le x_L$, set $t_1 = t^*$, $t_2 = \tilde{t}$, and go to step 17. Otherwise ($x(\tilde{t}) > x_L$), go to step 22.

12. [two reversals, v_0 and v_f negative, net movement to right] Same as step 9.

13. If $x^* < x_L$, set $t_1 = t_0$, $t_2 = t^*$, and go to step 17.

14. [first reversal occurs within current cell] If $x_R \le x(\tilde{t})$, set $t_1 = t^*$, $t_2 = \tilde{t}$, and go to step 16. Otherwise ($x(\tilde{t}) < x_R$), go to step 22.

15. [tracking for full time step stays within current cell] Set $t_0 = t_f$, t_f to the end of the next time step, $x_0 = x(t_0)$, $v_0 = v(x_0, t_0)$, $v_x = \dfrac{\partial v}{\partial x}(x_0, t_0)$, $v_t = \dfrac{\partial v}{\partial t}(x_0, t_0^+)$, and $v_{xt} = \dfrac{\partial^2 v}{\partial x \partial t}(x_0, t_0^+)$, where t_0^+ refers to the next time step that begins at t_0; x_L and x_R are unchanged. Go to step 23.

16. [tracking crosses right endpoint before end of time step] Use Newton/bisection with (8) to find the unique t', $t_1 < t' \le t_2$, such that $x(t') = x_R$. Set $t_0 = t'$, $x_0 = x(t') = x_R$, $v_0 = v(x_0, t_0)$, $v_x = \dfrac{\partial v}{\partial x}(x_0^+, t_0)$, $v_t = \dfrac{\partial v}{\partial t}(x_0, t_0)$, $v_{xt} = \dfrac{\partial^2 v}{\partial x \partial t}(x_0^+, t_0)$, $x_L = x_R$, and x_R to the right endpoint of the next cell to the right, where x_0^+ refers to the next cell to the right, whose left endpoint is x_0; t_f is unchanged. Go to step 23.

17. Like step 16, replacing right,x_R,x_0^+,x_L,left with left,x_L,x_0^-,x_R,right, respectively.

18. [tracking reaches right endpoint at end of time step] Set $t_0 = t_f$, $x_0 = x(t_0) = x_R$, $v_0 = v(x_0, t_0)$, $v_t = \dfrac{\partial v}{\partial t}(x_0, t_0^+)$, and t_f to the end of the next time step.

19. If $v_0 > 0$, or if $v_0 = 0$ and $v_t \ge 0$, set $v_x = \dfrac{\partial v}{\partial x}(x_0^+, t_0)$, $v_{xt} = \dfrac{\partial^2 v}{\partial x \partial t}(x_0^+, t_0^+)$, $x_L = x_R$, and x_R to the right endpoint of the next cell to the right. If $v_0 = 0$ and $v_t < 0$, set $v_x = \dfrac{\partial v}{\partial x}(x_0^-, t_0)$ and $v_{xt} = \dfrac{\partial^2 v}{\partial x \partial t}(x_0^-, t_0^+)$; x_L and x_R are unchanged. In either case, go to step 23.

20. Like step 18, replacing right,x_R with left,x_L.

21. Like step 19, replacing $>$,$\ge$,x_0^+,x_L,x_R,right,$<$, x_0^- with $<$,$\le$,x_0^-,x_R,x_L,left,$>$,x_0^+.

22. [track through the first of two reversals, staying in the current cell] Set $t_0 = \tilde{t}$, $x_0 = x(t_0)$, $v_0 = v(x_0, t_0)$, $v_x = \dfrac{\partial v}{\partial x}(x_0, t_0)$, and $v_t = \dfrac{\partial v}{\partial t}(x_0, t_0)$; v_{xt}, t_f, x_L, and x_R are unchanged. Go to step 23.

23. [one local tracking step completed] Go to step 1.

4 RESULTS AND DISCUSSION

4.1 *Example 1*

We first consider the one-dimensional problem presented in detail by Lu (1994). The simulated region is shown in Fig. 4 and consists of an aquifer of thickness 10 m that extends from a fully penetrating ditch at $x = 0$ to ∞. Aquifer hydraulic conductivity is .0002 m/s, specific storage is 2.0E-8 m^{-1}, and porosity is 0.5. Initially all heads in the aquifer and ditch are equal to 10 m. At time $t = 0$, the head in the ditch is instantaneously lowered to 0 and maintained at that level for all subsequent times. Velocities at any point in space and time can be determined from the analytical solution given by Carslaw & Jaeger (1959).

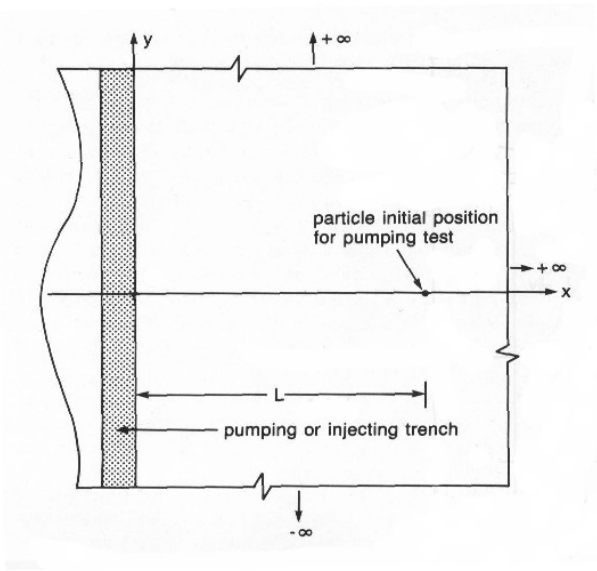

Figure 4: Aquifer, 10 m thick, simulated in example 1 (Lu 1994).

Table 1: Particle travel times (days) for example problem 1 with number of time steps NT = 15, injection time T (minutes) = 1 or 1000.

	Travel time	
Method	T=1	T=1000
Euler	14.35	21.15
present	13.28	20.78
Lu	13.28	20.77
Pollock	9.28	17.43

We used the same discretizations as Lu (1994): node spacings of 0.5 m and 15 time intervals (0.00035, 0.001, 0.01, 0.05, 0.2, 0.7, 1.2, 2.0, 3.0, 5.0, 9.0, 13.0, 17.0, 21.0, 30.0 days). Particles were tracked to the ditch from a distance of 5 m away for two initial times of release (1 and 1000 min).

Travel times between the injection point and the ditch are presented in Table 1 (NT = 15) for Euler integration with 1000 steps (Lu 1994), the method proposed here, the method of Lu (1994), and the semianalytical method of Pollock (1989) which assumed that within each time interval velocity was constant and equal to that at the end of the time interval. Values calculated for the latter two methods match those presented by Lu (1994) and thus imply that our computer program is correctly calculating travel times. The difference in results between our method and Lu's is trivial for this example and indicates that our method has no advantage. Indeed, the increased computational effort required by our method makes it less attractive than Lu's for this particular problem.

In order to test sensitivity of results to time discretization, the problem was rerun with 100 variably spaced time steps (NT = 100). Again we saw virtually no difference between our method and Lu's; however, results of all methods were closer to the Euler solution with the finer time discretization. In particular, the results of Pollock (1989) appeared more attractive, especially in light of the lesser computational requirements it has. A similar sensitivity was conducted on the spatial discretization; it showed very little change in results for refined

grids.

4.2 *Example 2*

For the second test problem, we modify the first example by extending to two dimensions and by allowing heterogeneity in the hydraulic conductivity field. The simulated domain now extends 5 m in the y-direction with spacing equal to that in the x-direction (0.5 m). The ditch extends across the entire length of the grid in the y-direction and initial and boundary conditions remain unchanged so that flow is still predominantly in the x-direction.

The natural log of hydraulic conductivity was randomly generated by the method of turning bands (Mantoglou & Wilson 1982). Spatial correlation was assumed over a distance of 3 m and was described by a spherical variogram. The mean and variance of the generated log values were -3.912 and 5, respectively. Specific storage was again set at 2.0E-8 m^{-1}. The flow equation was solved numerically with the grid extended in the x-direction to a distance sufficient to insure no boundary effects. The boundaries at $y = 0$ and 5 m were impermeable.

At an initial time of 200 sec, 100 particles were introduced at $x = 5$ m (10 particles equally spaced in each of the 10 cells in that column of the grid). Particles were tracked through the complex, 2-dimensional flow field until they reached the ditch. For each particle the starting and ending location and the total travel time were recorded. Results of different methods were compared on the basis of travel time and ending location. For each realization of the log hydraulic conductivity field, three simulations were made. These simulations differed only in the time discretization (number of time steps NT = 200, 100, or 20). Results from many simulations were studied and found to be qualitatively very similar. Therefore we present results from only two. To analyze results we consider the fine time discretization (NT = 200) using the present method to be the standard for comparison. This decision is somewhat justified by the fact that there was virtually no difference in results among the three methods for this discretization, and the present method accounts for the highest-order velocity variations. For each particle we compared the total travel time, t^f, and final y coordinate, y^f, to the values obtained from the standard solution (t_s, y_s, respectively). The mean square error (MSE) of travel time was calculated from normalized values of $(t_s - t^f)/t_s$. The average time deviation (DEV) and y-location deviation were calculated as average values of $|t_s - t^f|/t_s$ and $|y_s - y^f|/y_s$. Values for these terms are contained in Table 2.

The significance of the enhanced accuracy of the present method can be measured in two ways: by the differences between its results and those the other methods, and, when NT is decreased, by the amounts by which the errors of the other methods exceed those of the present method. For the fine time discretization (NT = 200), there is very little difference between results of our method and that of Lu (1994), while the Pollock (1989) method showed somewhat larger differences. As NT was decreased, the differences among all methods increased, and this trend toward differences of the order of 1% or more was apparent in all of the realizations examined. When NT changes, so does the computed velocity field; this causes the largest share of the errors in the bottom rows of the table. However, the present method appears to reduce the NT-sensitivity of the method of Lu (1994) by the order of 10% for this problem.

Table 2: Results of example 2 for number of time steps, NT, equal to 200, 100, 20.

	Realization 1				Realization 2			
Method	Average travel time	Time MSE	Time DEV	y DEV	Average travel time	Time MSE	Time DEV	y DEV
NT = 200								
present	8.98E+5	—	—	—	1.64E+5	—	—	—
Lu	8.98E+5	0.000	0.028	0.001	1.64E+5	0.000	0.002	0.001
Pollock	8.88E+5	0.002	0.036	0.000	1.60E+5	0.001	0.033	0.001
NT = 100								
present	8.93E+5	0.007	0.067	0.001	1.35E+5	0.013	0.090	0.001
Lu	8.92E+5	0.007	0.071	0.001	1.35E+5	0.014	0.094	0.002
Pollock	8.79E+5	0.015	0.120	0.001	1.28E+5	0.027	0.153	0.001
NT = 20								
present	9.24E+5	0.064	0.249	0.001	1.37E+5	0.028	0.165	0.002
Lu	9.09E+5	0.081	0.271	0.005	1.35E+5	0.032	0.177	0.005
Pollock	8.84E+5	0.110	0.325	0.001	1.26E+5	0.056	0.237	0.001

Table 3: Results of example 3 for number of time steps, NT, equal to 200, 100, 20.

	Realization 1				Realization 2			
Method	Average travel time	Time MSE	Time DEV	y DEV	Average travel time	Time MSE	Time DEV	y DEV
NT = 200								
present	2.87E+6	—	—	—	1.20E+7	—	—	—
Lu	2.93E+6	0.003	0.019	0.000	1.20E+7	0.000	0.005	0.001
Pollock	2.79E+6	0.001	0.034	0.000	1.18E+7	0.009	0.047	0.001
NT = 100								
present	2.60E+6	0.006	0.053	0.000	1.01E+7	0.013	0.080	0.000
Lu	2.62E+6	0.009	0.061	0.008	1.00E+7	0.014	0.081	0.002
Pollock	2.57E+6	0.008	0.069	0.007	9.93E+6	0.033	0.131	0.000
NT = 20								
present	2.87E+6	0.040	0.164	0.007	1.00E+7	0.024	0.134	0.002
Lu	2.14E+6	0.048	0.177	0.015	9.83E+6	0.032	0.147	0.005
Pollock	2.21E+6	0.061	0.201	0.020	1.02E+7	0.068	0.223	0.013

4.3 *Example 3*

The third test problem is identical to the second, except that ground-water velocities change direction during the simulation. Water level in the ditch is initially lowered to 0 m, then raised back up to 10 m, and then lowered to 0 m again. This causes the prominent direction of flow to be first toward, then away, and finally back toward the ditch. This example should produce relatively large values for $\partial^2 v/\partial x \partial t$ and should therefore be indicative of the largest differences to be expected between our method and that of Lu (1994). Results of these runs are shown in Table 3. The reduction of the NT-sensitivity is somewhat greater than in Example 2, and the differences in results in general, and particularly for the coarsest time discretization in Realization 1, are notably greater. In this regime, inclusion of the bilinear term can lead to highly significant differences in particle travel times.

5 CONCLUSIONS

The new particle-tracking method produces analytical trajectories in velocity fields that arise from a lowest-order Raviart-Thomas spatial discretization on a cartesian grid, fully accounting for linear temporal variations, including space-time bilinearity. Inclusion of the bilinearity is unimportant in some cases, but with spatial heterogeneity, and particularly with time-varying pumping or recharge, it can make large differences in travel times.

ACKNOWLEDGMENTS: The research of the first author was supported in part by National Science Foundation Grant No. DMS-9706866 and Army Research Office Grant No. 37119-GS-AAS.

REFERENCES

Carslaw, H. & Jaeger, J. (1959). *Conduction of Heat in Solids*. Oxford: Oxford University Press.

Garabedian, S. & Konikow, L. (1983). Front-tracking model for convective transport in flowing ground water. USGS Water Resources Investigations Report 83-4034.

Goode, D. (1990). Particle velocity interpolation in block-centered finite difference groundwater flow models. *Water Resour. Res. 26*(5), 925–940.

Healy, R. & Russell, T. (1993). A finite-volume Eulerian-Lagrangian localized adjoint method for solution of the advection-dispersion equation. *Water Resour. Res. 29*, 2399–2413.

Javandel, I., Doughty, L., & Tsang, C. (1984). *Groundwater Transport: Handbook of Mathematical Models*, Volume 10 of *Water Resources Monogr. Ser.* Washington, D.C.: AGU.

Konikow, L. & Bredehoeft, J. (1988). Computer model of two-dimensional solute transport and dispersion in ground water. In *Techniques of Water-Resources Investigations of the United States Geological Survey*, Volume 7, Washington. U.S. Government Printing Office.

Lu, N. (1994). A semianalytical method of path line computation for transient finite-difference groundwater flow models. *Water Resour. Res. 30*(8), 2449–2459.

Mantoglou, A. & Wilson, J. (1982). The turning bands method for simulation of random fields using line generation by a spectral method. *Water Resour. Res. 18*(5), 1379–1394.

Nelson, R. (1978). Evaluating the environmental consequences of groundwater contamination. *Water Resour. Res. 14*(3), 409–450.

Pollock, D. (1988). Semianalytical computation of path lines for finite-difference models. *Ground Water 26*(6), 743–750.

Pollock, D. (1989). Documentation of computer programs to compute and display pathlines using results from the U.S. Geological Survey modular three-dimensional finite difference groundwater flow model. U.S. Geol. Surv. Open File Rep. 89-381.

Prickett, T., Naymik, T., & Lonnquist, C. (1981). A "random-walk" solute transport model for selected groundwater quality evaluations. Bull. Ill. State Water Surv. 65.

Raviart, P. & Thomas, J. (1977). A mixed finite element method for 2nd order elliptic problems. In I. Galligani & E. Magenes (Eds.), *Mathematical Aspects of Finite Element Methods*, pp. 292–315. Springer-Verlag.

Russell, T. & Wheeler, M. (1983). Finite element and finite difference methods for continuous flows in porous media. In R. Ewing (Ed.), *The Mathematics of Reservoir Simulation*, Philadelphia, pp. 36–106. SIAM.

Schafer-Perini, A. & Wilson, J. (1991). Efficient and accurate front tracking for two-dimensional groundwater flow models. *Water Resour. Res. 27*, 1471–1485.

Shafer, J. (1987). Reverse pathline calculation of time-related capture zones in nonuniform flow. *Ground Water 25*(3), 283–289.

Zheng, C. (1989). PATH3D–a ground-water path and travel-time simulator. In *User's Manual*, Bethesda, Md. S.S. Papadopoulos and Associates.